CS 477/677 Analysis of Algorithms

Midterm Exam – Spring 2020

Student Name:

Section (circle the one that applies): CS 477 CS 677

- If you are in the CS 477 section you are required to answer problems 1 through 4, and if you are in section CS 677 you are required to answer problems 1 through 5. Problem 6 is a bonus question for everybody. If you are in section CS 477 you can solve either problem 5 or 6 as a bonus question (only the highest score of these two will be considered).
- Write your answers clearly and concisely. Do not re-prove theorems proved in the book or in class, but do justify their use clearly.
- If you need to exit the room and return during the test, you should let me know before leaving.
- This is a closed-book, closed-notes, closed-neighbor exam. Do not discuss or communicate with your fellow students on any of the questions.
- If you need extra paper, please let me know.

Question	Required	Points Received
1 (40 points)	All	
2 (15 points)	All	
3 (30 points)	All	
4 (15 points)	All	
5 (10 points)	Graduate/extra credit	
6 (10 points)	Extra credit	
Total		

1. [40 points]

(a) [15 points] Write pseudocode for a **recursive** algorithm that returns **the largest odd element** of an unordered array. If the array does not have any odd elements, the algorithm should return 0.

```
LO(A, p, r)
if p == r
    if A[p] is odd return A[p]
    else return 0
else
    m = ⌊(p+r)/2⌋
    left = LO(A, p, m)
    right = LO(A, m+1, r)
    return largest of left, right
```

(b) [15 points] Write a recurrence for the running time of the algorithm developed in part (a) and solve it using the method of your choice.

$$T(n) = 2T\left(\frac{n}{2}\right) + c$$

$f(n) = c$

$n^{\log_b a} = n$ $\Rightarrow$ Case 1 $\Rightarrow T(n) = \Theta(n)$

(c) [10 points] Compute the running time of the following algorithm (first find the exact formula and then give the running time in Θ notation). Show your entire computation for full credit.

ALGORITHM *Surprise*(A)

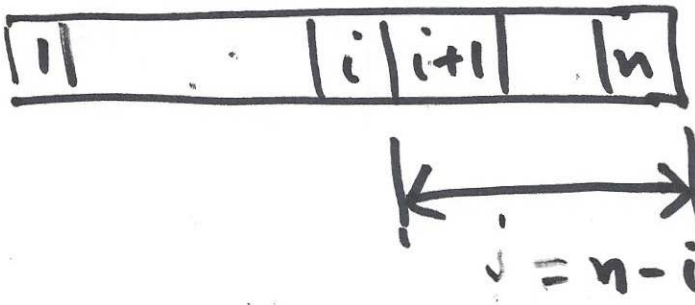
//Input: An array A[1..n] of integer numbers

for $i \leftarrow 1$ **to** n — $n+1$

do for $j \leftarrow n$ **downto** $i+1$ — $\sum_{i=1}^{n} (n-i+1)$

do if $A[j] < A[j-1]$

exchange $A[j]$ **with** $A[j-1]$ — $\sum_{i=1}^{n} (n-i)$



$j = n - i + 1$

$$T(n) = n+1 + \underbrace{\sum n}_{n^2} - \underbrace{\sum i}_{\frac{n(n+1)}{2}} + \underbrace{\sum 1}_{n} = \Theta(n^2)$$

2. [15 points] For each of the following statements indicate only whether it is true or false by circling True or False.

(a) True False

We can simultaneously find the minimum and the maximum numbers in an array using at most 3n/2 comparisons.

(b) True False

A max-heap is a reversely sorted array.

(c) True False

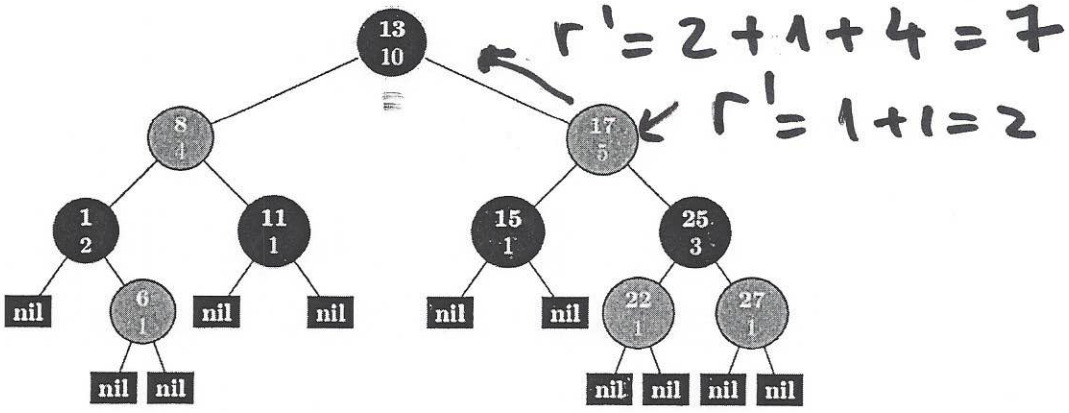
The non-randomized Partition algorithm always selects the smallest element of the array as a pivot.

(d) True False

The expected value of an indicator random variable associated with an event A is equal to the probability that event A will happen.

(e) True False

If ran on the tree below, OS_RANK ('root', pointer-to-node-17) will return 6.



3. [30 points] Answer the following questions:

(a) [8 points] For the **non-randomized** version of Quicksort, what are the best and the worst initial arrangements of the elements to be sorted? Justify your answer.

(b) [6 points] Arrange the following functions in ascending order of growth rate:

$f_1(n) = n^{2.5} \log n$

$f_2(n) = \sqrt[3]{2n^3 + n + 2}$

$f_3(n) = 3^n$

$f_4(n) = 3^2 = 9$

$f_5(n) = n^2 + n + 1$

$f_6(n) = 100n^{10} \frac{n \log n + n}{n^7} = n^3 \lg n \ldots$

$f_4 < f_2 < f_5 < f_1 < f_6 < f_3$

(c) [6 points] List and briefly describe the main steps used by the **divide and conquer** paradigm.

(d) [10 points] A bizarre weighted coin comes up heads with probability ½, tails with probability 1/3, and rests on its edge with probability 1/6. If it comes up heads, you win \$1. If it comes up tails, you win \$3. However, if it lands on its edge, you lose \$5. What is the expected winnings from flipping this coin?

$$E = \frac{1}{2} \cdot 1 + \frac{1}{3} \cdot 3 - \frac{1}{6} 5 = \frac{2}{3}$$

4. [15 points] Using mathematical induction, show that the following relation is true for every $n \geq 0$ and $x \neq 1$:

$$\sum_{i=0}^{n} x^i = \frac{x^{n+1} - 1}{x - 1}$$ TRUE

Base case : $n = 0$ $\quad x^0 = \frac{x^1 - 1}{x - 1}$ ✓

$1 = 1$

Assume : $\sum_{i=0}^{n+1} x^i = \frac{x^{n+2} - 1}{x - 1}$

Prove :

$$LHS = \frac{x^{n+1} - 1}{x - 1} + x^{n+1} = \frac{x^{n+1} - 1 + x^{n+2} - x^{n+1}}{x - 1} =$$

$$= \frac{x^{n+2} - 1}{x - 1}$$

5. [10 points] (Graduate/Extra credit problem)

Let **x** be a pointer to the root node of a red-black tree. Write an algorithm called **TreeCountRed(x)** that takes **x** as argument and returns **the number of red nodes** in the tree.

```
TCR(x)
{
    if (x == NIL)
        return ∅
    if  x is red
        return 1+ TCR(x.left)+TCR(
                                x.right)
    else
        return
            TCR(x.left)+TCR(x.right)
}
```

6. [10 points]. Extra Credit

Using the formal definition of the asymptotic notations, prove that $2n^3 + 100\lg n = O(n^4)$.

$\exists\ c, n_0 \quad 2n^3 + 100\lg n \leq c \cdot n^4$

$2n^3 + 100\lg n \leq 2n^3 + 100n^3 =$

$\lg n \leq n^3$

$= 102n^3 \leq 102 \cdot n^4.$

$n^3 < n^4 \qquad c = 102$